

Муниципальное бюджетное общеобразовательное учреждение
средняя общеобразовательная школа №18 имени В.М. Женко

Рассмотрена
на заседании
педагогического совета
протокол № 1 от
«30» 08 2022г

Согласована
Зам. директора
по ВР _____
«30» 08 2022 г.

Утверждена
Директором
МБОУ СОШ №18 _____
Приказ № «1»
01.09.2022г.

**РАБОЧАЯ ПРОГРАММА
ПЕДАГОГА ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ**

педагог: Алдошина Елена Анатольевна
название программы: **«Программирование на языке Python»**
направление: техническое
возраст обучающихся: 15-18 лет
срок реализации: 1 год

г. Узловая
2022 год

Пояснительная записка

Актуальность

Информатика в настоящее время - стремительно развивающаяся область практической деятельности человека, связанная с использованием компьютерных технологий. В современных условиях образовательная деятельность в этой сфере является чрезвычайно востребованной.

Ключевой особенностью курса является его направленность на формирование у учащихся навыков поиска собственного решения поставленной задачи, составления алгоритма решения и реализации алгоритма с помощью средств программирования.

В рамках предлагаемого курса «Программирование на Питоне» - это не столько средство подготовки к будущей профессиональной деятельности, сколько формирование новых общеинтеллектуальных умений и навыков: разделение задачи на этапы решения.

Направленность - техническая.

Адресат программы - обучающиеся 15-18 лет, увлеченные компьютерным программированием.

Объем и срок реализации программы - **1 год, 70 часов.**

Цель программы:

- понять значение алгоритмизации как метода познания окружающего мира, принципы структурной алгоритмизации;
- овладеть базовыми понятиями теории алгоритмов;
- научиться разрабатывать эффективные алгоритмы и реализовывать их в виде программы, написанной на языке программирования Python.

Задачи курса:

- познакомить с понятиями алгоритма, вычислимой функции, языка программирования;
- научить составлять и читать блок-схемы;
- сформировать навыки выполнения технологической цепочки разработки программ средствами языка программирования Python;
- изучить основные конструкции языка программирования Python, позволяющие работать с простыми и составными типами данных (строками, списками, кортежами, словарями, множествами);
- научить применять функции при написании программ на языке программирования Python;
- научить отлаживать и тестировать программы, делать выводы о работе этих программ.

Методы обучения

Отбор методов обучения обусловлен необходимостью формировать информационную и коммуникативную компетентности учащихся, реализовывать личностно-ориентированное обучение, направлять их на самостоятельное решение разнообразных проблем, развивать исследовательские и творческие способности. Решение данных задач кроется в организации деятельностного подхода к обучению, в проблемном изложении материала учителем, в переходе от репродуктивного вида работ к самостоятельным, поисково-исследовательским видам деятельности. Поэтому основная методическая установка в данном курсе — обучение учащихся навыкам самостоятельной творческой деятельности.

Формы организации учебных занятий

Организация учебного процесса предусматривает дистанционной формы деятельности, когда учащийся вне уроков самостоятельно выполняет на компьютере практические задания.

Планируемые результаты курса

В рамках курс «Языки программирования. Python» учащиеся овладевают следующими знаниями, умениями и способами деятельности:

- умеют составлять алгоритмы для решения задач;
- умеют реализовывать алгоритмы на компьютере в виде программ, написанных на языке Python;
- владеют основными навыками программирования на языке Python;
- умеют отлаживать и тестировать программы, написанные на языке Python.

Способы оценивания уровня достижений учащихся

Предметом диагностики и контроля в курсе «Основы программирования на языке Python» являются внешние образовательные продукты учащихся (созданные блок-схемы, программы), а также их внутренние личностные качества (освоенные способы деятельности, знания, умения), которые относятся к целям и задачам курса.

Качество внешней образовательной продукции желательно оценивать по следующим параметрам:

- алгоритм должен быть оптимальным по скорости выполнения и максимально простым в реализации на языке программирования;
- программа должна выполнять поставленные задачи;
- по степени «читаемости кода» (должны быть соблюдены отступы, обязательное наличие комментариев к коду программы и т. д.).

Проверка достигаемых учащимися результатов производится в следующих формах:

- текущий рефлексивный самоанализ, контроль и самооценка учащимися выполняемых заданий;
- текущая диагностика и оценка учителем деятельности школьников;
- итоговая оценка деятельности и образовательной продукции ученика в соответствии с его индивидуальной образовательной программой освоения курса;

Итоговый контроль проводится в конце всего курса. Он организуется тестирования.

Программное обеспечение:

1. Операционная система: Windows 7.0 (или выше).
2. Среда разработки: Python 3.3 (или выше),

Учебно-тематический план

№ п/п	Наименование разделов и тем	Всего часов	Количество часов	
			Теоретические	Практические
1.	Знакомство с языком Python	4	2	2
2.	Переменные и выражения	8	4	4
3.	Условные выражения	12	4	8
4.	Циклы	18	8	10
5.	Функции	12	4	8
6.	Строки - последовательности символов	6	3	3
7.	Сложные типы данных	9	4	5
8.	Стиль программирования и отладка программ	2	1	1
	ИТОГО	70	30	40

Содержание тем учебного курса

Тема 1. Знакомство с языком Python

Общие сведения о языке Python. Установка Python на компьютер. Режимы работы Python.

Что такое программа. Первая программа. Структура программы на языке Python. Комментарии.

Практические работы:

Установка программы Python

Режимы работы с Python

Учащиеся должны знать / понимать:

- понятие программы;
- структура программы на Python;
- режимы работы с Python.

Учащиеся должны уметь:

- выполнить установку программы;
- выполнить простейшую программу в интерактивной среде;
- написать комментарии в программе.

Тема 2. Переменные и выражения

Типы данных. Преобразование типов. Переменные. Оператор присваивания. Имена переменных и ключевые слова.

Выражения. Операции. Порядок выполнения операций. Математические функции. Композиция.

Ввод и вывод. Ввод данных с клавиатуры. Вывод данных на экран. Пример скрипта, использующего ввод и вывод данных. Задачи на элементарные действия с числами. Решение задач на элементарные действия с числами.

Практические работы:

Работа со справочной системой.

Переменные.

Выражения.

Задачи на элементарные действия с числами.

Учащиеся должны знать / понимать:

- общую структуру программы;
- типы данных;
- целые, вещественные типы данных и операции над ними;
- оператор присваивания;
- операторы ввода-вывода.

Учащиеся должны уметь:

- пользоваться интерфейсом среды программирования Python;
- использовать команды редактора;
- организовывать ввод и вывод данных;
- записывать арифметические выражения.

Тема 3. Условные предложения

Логический тип данных. Логические выражения и операторы. Сложные условные выражения (логические операции and, or, not). Условный оператор. Альтернативное выполнение. Примеры решения задач с условным оператором. Множественное ветвление. Реализация ветвления в языке Python.

Практические работы:

Логические выражения.

Условный оператор.

Множественное ветвление.

Условные операторы.

Учащиеся должны знать / понимать:

- назначение условного оператора;
- способ записи условного оператора;

- логический тип данных;
- логические операторы or, and, not;

Учащиеся должны уметь:

- использовать условный оператор;
- создавать сложные условия с помощью логических операторов.

Тема 4. Циклы

Понятие цикла. Тело цикла. Условия выполнения тела цикла. Оператор цикла с условием. Оператор цикла while. Бесконечные циклы. Альтернативная ветка цикла while. Обновление переменной. Краткая форма записи обновления. Примеры использования циклов.

Оператор цикла с параметром for. Операторы управления циклом. Пример задачи с использованием цикла for. Вложенные циклы. Циклы в циклах. Случайные числа. Функция randrange. Функция random. Примеры решения задач с циклом.

Практические работы:

Числа Фибоначчи.

Решение задачи с циклом for.

Реализация циклических алгоритмов.

Случайные числа.

Решение задач с циклом.

Учащиеся должны знать / понимать:

- циклы с условием и их виды;
- правила записи циклов условием;
- назначение и особенности использования цикла с параметром;
- формат записи цикла с параметром;
- примеры использования циклов различных типов.

Учащиеся должны уметь:

- определять вид цикла, наиболее удобный для решения поставленной задачи;
- использовать цикл с условием;
- определять целесообразность применения и использовать цикл с параметром для решения поставленной задачи;

Тема 5. Функции

Создание функций. Параметры и аргументы. Локальные и глобальные переменные.

Поток выполнения. Функции, возвращающие результат. Анонимные функции, инструкция lambda. Примеры решения задач с использованием функций.

Рекурсивные функции. Вычисление факториала. Числа Фибоначчи.

Практические работы:

Создание функций.

Локальные переменные.

Решение задач с использованием функций

Рекурсивные функции

Учащиеся должны знать / понимать:

- понятие функции;
- способы описания функции;
- принципы структурного программирования;
- понятие локальных переменных подпрограмм;
- понятие формальных и фактических параметров подпрограмм;
- способ передачи параметров.

Учащиеся должны уметь:

- создавать и использовать функции;
- использовать механизм параметров для передачи значений.

Тема 6. Строки - последовательности символов

Составной тип данных - строка. Доступ по индексу. Длина строки и отрицательные индексы. Преобразование типов. Применение цикла для обхода строки.

Срезы строк. Строки нельзя изменить. Сравнение строк. Оператор in. Модуль string. Операторы для всех типов последовательностей (строки, списки, кортежи). Примеры решения задач со строками.

Практические работы:

Строки.

Решение задач со строками.

Учащиеся должны знать / понимать:

- назначение строкового типа данных;
- операторы для работы со строками;
- процедуры и функции для работы со строками;
- операции со строками.

Учащиеся должны уметь:

- описывать строки;
- соединять строки;

- находить длину строки;
- вырезать часть строки;
- находить подстроку в строке;
- находить количество слов в строке.

Тема 7. Сложные типы данных

Списки. Тип список (list). Индексы. Обход списка. Проверка вхождения в список.

Добавление в список. Суммирование или изменение списка. Операторы для списков.

Срезы списков. Удаление списка. Клонирование списков. Списочные параметры.

Функция range. Списки: примеры решения задач.

Матрицы. Вложенные списки. Матрицы. Строки и списки. Генераторы списков в Python. Кортежи. Присваивание кортежей. Кортежи как возвращаемые значения. Введение в словари. Тип словарь (dict). Словарные операции. Словарные методы.

Множества в языке Python. Множества. Множественный тип данных. Описание множеств. Операции, допустимые над множествами: объединение, пересечение, разность, включение. Оператор определения принадлежности элемента множеству.

Практические работы:

Списки.

Решение задач со списками.

Учащиеся должны знать / понимать:

- сложные типы данных;
- способ описания списка;
- способ доступа к элементам списка;
- способ описания кортежа;
- способ описания словаря;
- операции, выполняемые со списками, кортежами и словарями;
- понятие множества;
- способы описания множества;
- операторы работы с множествами.

Учащиеся должны уметь:

- описывать списки;
- вводить элементы списка;

- выводить элементы списка;
- выполнять поиск элемента в списке, поиск минимума и максимума, нахождение суммы элементов списка;
- использовать вложенные списки;
- приводить примеры использования вложенных списков (матриц);
- описывать множества;
- определять принадлежность элемента множеству;
- вводить элементы множества;
- выводить элементы множества.

Тема 8. Стиль программирования и отладка программ

Стиль программирования. Отладка программ.

Учащиеся должны знать / понимать:

- что такое стиль программирования;
- правила именования объектов;
- основные рекомендации при написании программ.

Учащиеся должны уметь:

- определять вид ошибок и находить ошибки в программе.
- выполнять тестирование и отладку программ.

Литература и источники

1. Домашняя страница Python www.python.org . Справочные материалы, официальная документация.
2. Сайт проекта Интуит: Национальный открытый университет, курс «Введение в программирование на Python», <http://www.intuit.ru/studies/courses/12179/1172/info>.
3. Сайт проекта Интуит: Национальный открытый университет. Курс «Язык программирования Python» <http://www.intuit.ru/studies/courses/49/49/info>.
4. Сайт проекта Open Book Project openbookproject.net содержит серию практических примеров на Python Криса Мейерса.
5. *Python. Подробный справочник* Дэвида М. Бизли — книга со справочной информацией о языке Python и модулях стандартной библиотеки.
6. *Python. Справочник* Марка Лутца. Справочник по наиболее часто используемым функциям и модулям.

Календарно-тематического планирование
кружка «Программирование на языке Python»
 на 2022-2023 учебный год, 52 часа

№ занятия	Наименование разделов и тем	Кол-во часов	Планируемые сроки	Фактические сроки
	Знакомство с языком Python	4		
1	Общие сведения о языке.	1	05.09.22	
2	Практическая работа. Установка программы Python	1	05.09.22	
3	Режимы работы.	1	12.09.22	
4	Практическая работа. Режимы работы с Python	1	12.09.22	
	Переменные и выражения	8		
5	Переменные	1	19.09.22	
6	Практическая работа. Работа со справочной системой	1	19.09.22	
7	Практическая работа. Переменные	1	26.09.22	
8	Выражения.	1	26.09.22	
9	Практическая работа. Выражения	1	03.10.22	
10	Ввод и вывод данных	1	03.10.22	
11	Задачи на элементарные действия с числами	1	10.10.22	
12	Практическая работа. Задачи на элементарные действия с числами.	1	10.10.22	
	Условные предложения	12		
13	Логические выражения и операторы	1	17.10.22	
14	Практическая работа. Логические выражения	1	17.10.22	
15	Условный оператор.	1	24.10.22	

16-17	Практическая работа. "Условный оператор"	2	31.10.22	
18	Множественное ветвление	1	07.11.22	
19-20	Практическая работа. "Множественное ветвление".	2	14.11.22	
21	Реализация ветвления в языке Python.	1	21.11.22	
22-24	Практическая работа. "Условные операторы"	3	21.11.22 28.11.22	
	Циклы	18		
25	Оператор цикла с условием	1	05.12.22	
26	Практическая работа. "Числа Фибоначчи"	1	05.12.22	
27	Оператор цикла for	1	12.12.22	
28-29	Практическая работа. Решение задач с циклом for.	2	12.12.22	
30-31	Вложенные циклы	2	19.12.22	
32-34	Практическая работа. Реализация циклических алгоритмов	3	26.12.22	
35	Случайные числа	1	09.01.23	
36-37	Практическая работа. Случайные числа	2	09.01.23	
38-39	Примеры решения задач с циклом	2	16.01.23	
40-42	Практическая работа. Решение задач с циклом.	3	16.01.23	
	Функции	12		
43	Создание функций	1	23.01.23	
44-45	Практическая работа. Создание функций	2	30.01.23	
46	Локальные переменные	1	06.02.23	
47-48	Практическая работа. Локальные переменные	2	13.02.23	
49	Примеры решения задач с использованием функций	1	20.02.23	

50-51	Практическая работа. Решение задач с использованием функций	2	27.02.23	
52	Рекурсивные функции	1	06.02.23	
53-54	Практическая работа №19. Рекурсивные функции	2	13.03.23	
	Строки - последовательности символов	6		
55	Строки	1	20.03.23	
56	Срезы строк	1	20.03.23	
57-58	Практическая работа №20. Строки	2	27.03.23	
59	Примеры решения задач со строками	1	03.04.23	
60	Практическая работа №21. Решение задач со строками.	1	03.04.23	
	Сложные типы данных	9		
61	Списки	1	10.04.23	
62	Срезы списков	1	10.04.23	
63	Практическая работа №22. Списки	1	17.04.23	
64	Списки: примеры решения задач	1	17.04.23	
65	Практическая работа №23. Решение задач со списками	1	24.04.23	
66	Матрицы	1	15.04.23	
67	Кортежи	1	15.04.23	
68	Введение в словари	1	22.05.23	
69	Множества в языке Python	1	22.05.23	
	Стиль программирования и отладка программ	2		
70	Стиль программирования	1	29.05.23	
71	Отладка программ	1	29.05.23	